

6/PRTS

An Unique versatile executor engine which can interpret and execute transaction structures and information views to build information systems.

#### Field of Invention

This invention in general relates to the field of building information systems for enterprises. Further, this invention relates to an unique versatile executor engine, which can interpret and execute transaction structures and information views. Hence, the information system thus built can grow along with the needs of the enterprise.

The novelty lies in the new methodology that has been evolved to build information systems with an executor engine.

An information system built in this methodology will allow online and on site creation of new options and changing the existing options without any down time.

Software programs are not written to build an information system in this methodology. The information system is analyzed and split into transaction structures and information views. These are then defined using the PROFIT 5RM builder, which stores these definitions as data. This is interpreted and executed by the executor engine to provide the necessary information system. The architecture of the system is described in detail and illustrated in the first figure of the drawing.

Accordingly an information system can be built in very short time. An estimated minimum of 1:50 time saving can be achieved in this methodology in comparison with the prior art.

Further the system enables domain experts to build information systems without the help of programmers or programming team.

DT04 Rec'd PCT/PTO 0 8 JUL 2004

The salient feature of the invention is in the provision of features to overcome the following limitations in the prior art.

The steps involved to provide an end to end business solution are various and elaborate, often time consuming and developed at huge costs.

At first the consumer poses his requirements to the domain expert who in turn analyses and structures the solution and passes it on to the programming team. They then write a program to suit the needs. Such programs will be to a specific process and have limited flexibility. Any further changes, if at all to be incorporated, have to be made by the programming team.

The success of this novel program according to the invention depends upon the ability of the programming team, to comprehend fully both the business process in its entirety and the technical know how. Such people quipped with both domain expertise and technical expertise are very rare in the industry. So, sourcing such people, hiring the team and the time constraint could only send the developing costs, spiraling up, rendering such solution providers less competitive.

#### Prior art

It is to be noted that none of the existing business process solution providers has an executor interpreter that can interpret transaction structures and information views. So, programming skills are essential to maneuver the software. This results in considerable investment of human resources, hardware, and software resources.

It is obvious that the existing business process solution providers target specific business processes limiting their scope of usage and also their flexibility to accommodate changes. Because of this flexibility constraint, such solution providers cannot grow with the needs on time.

The architecture is not adaptable to change in business trends and very few options are available to customize and often become redundant. The objects of handling core functionality are huge and hence in a significant measure render them less scalable.

The code keeps on changing from customer to customer making them more bug-prone. The domain expert cannot build solutions without the help of programming team.

Huge programming time, less flexibility, considerable maintenance costs (programmers to being involved at each stage of accommodating change), high failure rate, lack of versatility (provide solutions to specific business needs) makes it very difficult to build and maintain information systems.

It is envisaged to invent an "unique executor engine" which encompasses and caters to each and every requirement of business process. It is versatile, flexible, substantial time and cost-saving, user friendly, easily scalable, no programming skills required and all importantly no failure and never redundant.

Accordingly it is the primary object of the invention to invent an unique versatile executor engine, which interprets and executes the transaction structures and information views.

It is another object of the invention to invent a new methodology, which simplifies business software development.

It is another object of the invention to invent an unique executor engine, which is versatile and flexible.

It is another object of the invention to invent an unique versatile executor engine which on use results in substantial time saving.

It is another object of the invention to invent an unique versatile executor engine, which on use results in substantial time saving.

It is another object of the invention to invent an unique versatile executor engine wherein no programming skills are required.

Further objects of the invention will be clear from the following description:

An unique versatile executor engine which interprets transaction structures and information views wherein the said software tool, has two parts namely builder and executor engine. The builder will capture the business process in terms of transaction structures with process maps and information views. The executor engine will read and execute the definitions. The architecture of the executor engine has been made in such a way that a way that scaling to new technologies can be made with ease without affecting the existing information system.

Accordingly the invention has many essential features which are enumerated in the following description.

1. An unique versatile executor engine which can interpret and execute transaction structures and information views to build information system comprises essentially a software tool being of two parts, a builder and an executor engine, the builder being to capture the business process in terms of transaction structure with process maps and information views, the executor engine being to read and execute the definitions, the said executor engine having an unique architecture in such a way that scaling of new technologies can be made with ease without affecting the existing information system.

2. An unique versatile executor engine which can interpret and execute transaction structures and information views to build information system wherein the transaction structure is a collection of
  - a) Data containers
  - b) GUI Definitions
  - c) Validation statements
  - d) Process maps
  - e) Print formats
3. An unique versatile executor engine which can interpret and execute transaction structures and information views to build information system wherein the said executor engine has an unique technology, which enables an information system to be built by representing the system requirements as transaction structures and information views and storing them as data, instead of writing software programs to achieve the same, thus making information system building simple and quick.
4. An unique versatile executor engine which can interpret and execute transaction structures and information views to build information system wherein the said executor engine has open-ended architecture of PROFIT 5RM which facilitates easy scalability to new technologies without affecting the already made information structure and further, allows change in the information structure immaterial of the technology extensions made.
5. An unique versatile executor engine which can interpret and execute transaction structures and information views to build information system wherein PROFIT 5RM incorporates an a unique technology, which enables an information system to be built without writing software programs and this makes information system building simple and quick.

6. An unique versatile executor engine which can interpret and execute transaction structures and information views to build information system wherein PROFIT 5RM has the technology in which the information system is analyzed and split into transaction structures and information views and these are then defined using the PROFIT 5RM builder, which stores these definitions as data and this being interpreted and executed by the unique executor engine to provide the necessary information.
7. An unique versatile executor engine which can interpret and execute transaction structures and information views to build information system wherein application of software reduces software development cycle time in a ratio of 1:50 as compared to the traditional methodology.
8. An unique versatile executor engine which can interpret and execute transaction structures and information views to build information system wherein the said executor engine possesses a technology which is domain-neutral and hence can be used to build information systems for all domains.
9. An unique versatile executor engine which can interpret and execute transaction structures and information views to build information system wherein the system possesses a technology, which facilitates onsite as well as online development and which can be maintained with ease.

The nature of the invention and the manner in which the invention is to be performed is clearly described through drawings, which accompany the description.

#### **SHORT DESCRIPTION OF FIGURES OF DRAWINGS**

Fig.1 of the drawing shows the open-ended architecture of PROFIT 5RM

Fig.2 of the drawing shows the development cycle according to prior art.

Fig.3 of the drawings shows the development cycle according to invention.

Fig.4 of the drawings shows the time frame for building an information system in a conventional method.

Fig.5 of the drawings shows the time frame for building an information system with products developed in the prior art.

Fig.6 of the drawings shows the time frame for building an information system using PROFIT 5 RM.

## **DÉTAILED DESCRIPTION OF A PREFERRED EMBODIMENT OF THE INVENTION**

### **PROFIT 5RM ARCHITECTURE**

Refer to Figure 1 for the open ended architecture diagram of PROFIT 5RM.

A domain expert interacts with the end user and understands the information system required by the organization.

The information system is analyzed and made into simple structures called Transaction Structures and Information Views. These are defined using the PROFIT 5RM Builder.

The Builder stores the definitions of Transaction Structures and Information View as data into the database.

The Graphical User Interface Layer (GUIL) presents a user interface for the end user based on the Transaction Structure.

The GUIL interacts with the end user and submits various requests for transaction processing or information request to the Process Request Server [PRS].

Process Request Server(PRS) provides services to process transactions and information requests. The **Transaction Structures** govern the transaction processing done by the PRS. The **Information Views** govern the processing of information requests.

In this approach, software programs are not written to make an information system. Instead, an information system is made into Transaction Structures and Information View and stored in the database.

#### **Transaction Structures[Tstruct]**

Transaction Structures are the building blocks of an information system. These govern the data input, validation and storage functionality of an information system.

A transaction structure is collection of

- Data containers
- GUI Definitions
- Validation statement
- Process maps
- Print formats

The executor engine interprets a Transaction Structure and a transaction form is presented to the end user. The end user can input data into the information system through these forms.

**Data Containers** are logical groups of input data in transaction structure. Containers can be classified into simple and tabular. A simple container will hold input fields which will hold only one value in a single transaction. A tabular container will hold input fields in the form of the grid and hence can hold many values in single transaction. A transaction structure and have any number of containers.



**Input fields** are elements, which hold a piece of data of a transaction. Each container can have one or more input fields.

An input field is identified through a name that is unique in a transaction structure.

Input fields can be categorized into simple, calculated, auto - generated and referral fields.

Simple input fields accept data into it based on the type definition.

Calculated input fields, evaluate an expression to get its value. The expression could be based on any field in the Tstruct.

Auto - generated fields generate serial number for each transaction.

Referral fields allow user to pick data from other transactions. Defining SQL statement, as part of the input field does this. The result set of the SQL will be presented to the end user for selection.

GUI Definitions can be made to set the look and feel of the transaction forms.

The Validation Statement is an expression, which controls the validity of each transaction entered through the transaction form.

Process maps provide a way of relating transaction structures and passing data from one transaction to another.

Three kinds of process maps are required for an information system:

- Transaction Generator

- Master - Detail
- Transaction - Group

**Transaction Generator** maps a source transaction structure to a target transaction structure. Such a mapping will create one or more target transaction automatically when the source transaction is stored. Mapping fields in the source to that in the target does it.

**Master - Detail** maps are made to update data from a detail transaction into another related transaction called the master transaction. The field in the details is mapped to a field in the master along with the type of update to be done.

**Transaction - Group** maps represent a many - to - many user - controlled relationship between transactions from many transaction structures. Transactions from the different structures that need to be related are posted to a common transaction structure. This common transaction structure is called the **Group Structure**.

Each group is a collection of transactions. The first transaction in the group is called the **group header** and the subsequent transactions are called the group elements. An input field in the group Validation Field (GVF). A group is valid only if the value in the GVF in the group header is greater than or equal to the sum of the GVF values in the group elements.

**Print formats** are text documents with input field embedded in it. A transaction form can be printed along with the data entered in it. The format of the printed document will be the same as the print format with the input fields replaced with actual data.

### INFORMATION VIEW I View

An Information View is a set of related SQL statements that provide data to a view format. It governs the information presentation function of an information system.

Information view are of two types

- Tabular
- Free form

The classification is based on the format in which the view is to be presented. ....

A Tabular View consists of a set of SQL statements. The SQL result sets are related by defining a relation field. Rows in the result sets that have the same value in the relation field are collected into one row & presented in the IView.

Every column in each result set is represented as column in the IView. The IView can have additional columns which may be computed based on an expression. The expression may include conditional statements also.

Columns in the Iview can be grouped based on a GROUP FIELD. All rows that have the same value in the group field are collated & presented as one row in the IView.

Multi level control breaks can be defined in a an IView by defining key columns. A total line is printed when the value in the key column change. Further, report balance can be defined on any control break. In such a case, a closing and opening balance is automatically printed.

The features discussed above, facilitate making of all kinds of tabular reports that may be required in an information system without writing programs to achieve the same.

A **Free Form View** consists of a set of SQL statements and a text document with print fields embedded in the document.

Print fields are replaced with actual data before printing the report. The actual data are drawn from the SQL result sets.

The executor engine reads an Iview, executes the SQL statements and prepares the Iview and presents it as an information form to the user.

The Information form provides features to navigate through the Iview and select any row for further drill down.

A drill down facility can be achieved by linking two Iview. Any column from the selected row can be set to provide parameter values to the SQL statements in the linked Iview.

A row in the Iview may also be linked to a transaction form. In such a case, the transaction form will be displayed by the executor engine with data related to the selected row in the Iview.

### Charts

Figure .1 of the drawing shows the open-ended architecture of PROFIT 5RRM.

Figure. 2 of the drawings shows the development cycle as per the prior art, which includes requirement study, system design, database design, coding, testing, training and implementation.

The software executor engine invented by us will cut short the software development cycle to a great extent. Programming efforts will be removed completely from the development cycle. The time saving could be described as follows: What can be done by an experienced programmer in a month, can be done by a non-programmer in just a

few hours. The time saving ratio is 1:50. Figure 3 of the drawings shows the development cycle with PROFIT 5 RM.

Figure 4 shows the time frame for building an information system in conventional method. The methodology includes development of the modules for a small manufacturing enterprise right from system study, design & complete coding. The basic modules covered are financial accounting, inventory, production planning and control, order processing and payroll.

Figure 5 shows in detail the time frame for building an information system with products developed in the prior art. This methodology includes customization of basic existing modules for a small manufacturing enterprise. The enhancements could be achieved only by modifying the program. The basic modules covered are financial accounting, inventory, production planning and control, order processing and payroll.

Figure 6 shows the time frame for building an information system using PROFIT 5RM. This methodology includes definition of the complete requirements for a small manufacturing enterprise without programming. The enhancements could be achieved by merely modifying the definitions but not coding. The basic modules covered are financial accounting, inventory, production planning and control, order processing and payroll.

The invention provides a new method to build and create an information system.

## **Advantages of PROFIT 5RM architecture**

### **Domain Neutral**

Information systems for any kind of enterprise, industry or organization can be built using Profit 5RM.

### **Time**

This methodology completely eliminates software programming. Hence, an information system can be built in a very short time and with ease.

### **Cost**

A domain expert can build an information system without using a programming team and hence, the built information system will cost much less.

### **Maintenance**

The transaction structures and information views that are the building blocks of the information system are stored as data and hence can be maintained with ease.

### **Change Management**

Changes to processes can be done on the fly and at site because changes to processes now mean changes to transaction structures and information views that are stored as data.

### **No Programming Required**

Domain experts need not depend on a programming team get an information system in place with this methodology as there is no programming involved. Hence a solution built on this platform will have an extremely high rate of success.

### **Easy Scalability**

The PRS services are simple objects, which can be extended to support new technologies by wrapping them in appropriate interface classes. For example these services may be

made into Web services by creating SOAP (simple object Access Protocol) interfaces. Hence, this layer can be scaled to any new technology with ease and speed.

As business processes are stored as data, scaling the PRS will not affect the information system that has been already defined.

Similarly, altering the definitions immaterial of the technology extensions made to the PRS can change the business process themselves.

The architecture facilitates PROFIT SRM to act as an information system foundation for any enterprise.

It is therefore evident that the Application of the invention is in building information systems for all lines of businesses.

#### **Salient Features of the Invention**

1. A claim is made for the incorporation of a unique technology, which enables an information system to be built by representing the system requirements as Transaction structures and information views and storing them as data, instead of writing software programs to achieve the same, thus making information system building simple and quick.
2. A claim is made for the incorporation of an open-ended architecture of PROFIT SRM (as illustrated in Fig.1). This facilitates easy scalability to new technologies without affecting the already made information structure. Further, the technology allows change in the information structure immaterial of the technology extensions made.
3. A claim is made for the unique versatile executor engine, which interprets and executes the transaction structures and information views.

4. A claim is made for the technology, which reduces software development cycle time in a ratio of 1:50 compared to the traditional methodology.
5. A claim is made for the technology, which is domain-neutral and hence that can be used to build information systems for all domains.
6. A claim is made for the technology, which facilitates onsite as well as online development and which can be maintained with ease.